UNIVERSITA' DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione

Organizzazioni virtuali di tipo Peer-to-Peer

Ing. Michele Amoretti
Distributed Systems Group

VI Incontro del GARR
17 Novembre 2005

Sommario

- Organizzazioni virtuali

- Modelli architetturali P2P

- Skype

Organizzazioni virtuali

Definizione e requisiti

Una organizzazione virtuale è una comunità formata da individui e istituzioni, che sono *nodi attivi della rete* perchè condividono informazioni, risorse e servizi.

L'infrastruttura deve garantire:
- replicazione delle risorse
- reperibilità delle risorse
- sicurezza

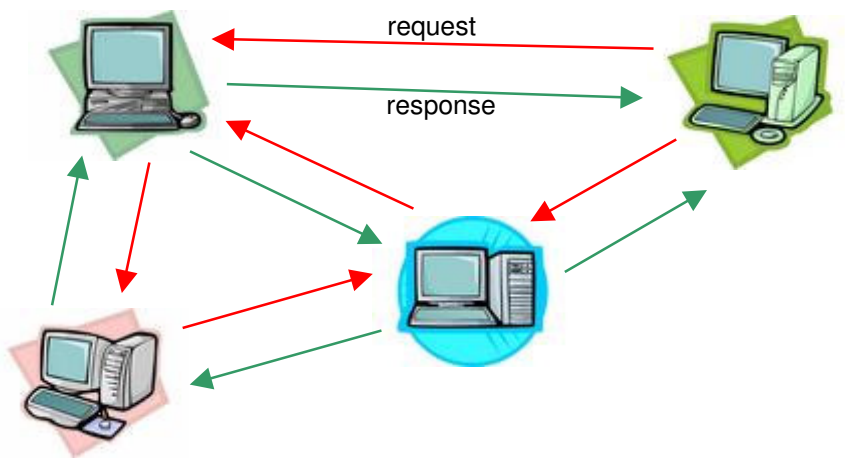
e deve evitare:
- contese
- inutilizzo di alcune risorse, ed eccessivo utilizzo di altre

Stato dell'arte: *tecnologie Grid*.

Paradigmi: *Client/Server*, ***Peer-to-Peer***.

Sistema Peer-to-Peer

In un sistema P2P ciascun nodo ha funzionalità sia di client che di server, e può essere parzialmente o completamente autonomo nel senso che non dipende da alcuna autorità centrale.



Organizzazioni virtuali di tipo P2P

Infrastrutture P2P per costituire e connettere comunità omogenee o eterogenee:

Campus Universitari

Laboratori di Ricerca

Aziende

Filiere

Modelli architetturali P2P

Modello architetturale P2P = rete di overlay + algoritmo di routing

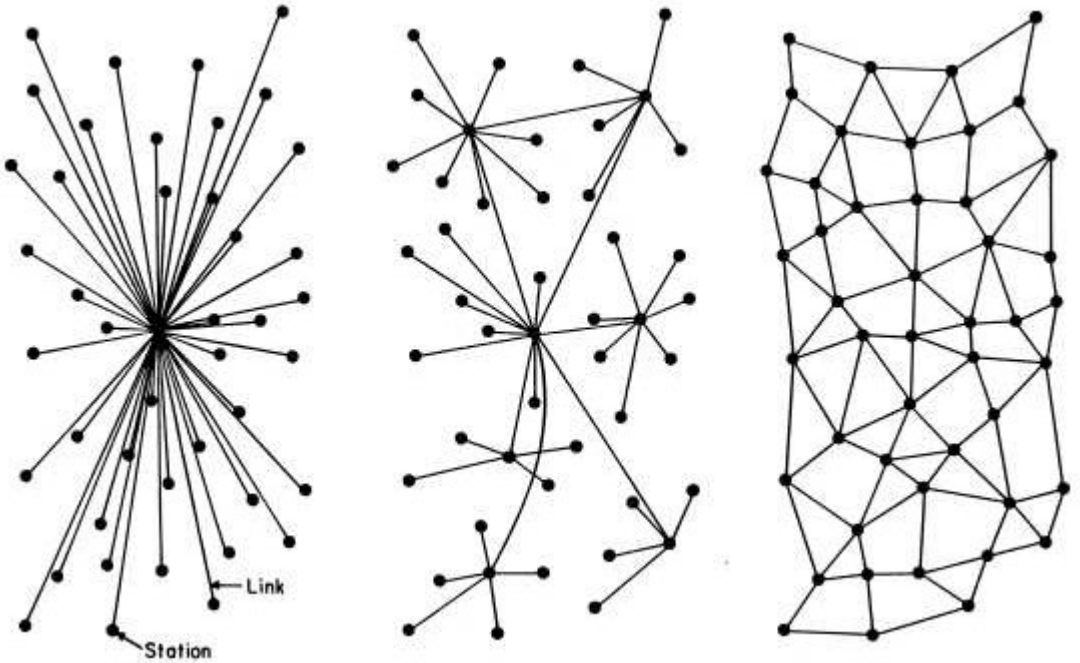
Un sistema P2P è scalabile se l'architettura della rete di overlay (definita da una *topologia* e dalla presenza o assenza di *struttura logica*) contribuisce a rendere efficiente l'algoritmo di routing.

Quest'ultimo, in fase di ricerca, deve instradare i messaggi nella direzione più opportuna, coprendo il percorso più breve possibile, e minimizzando l'overhead computazionale nei nodi e la duplicazione dei messaggi.

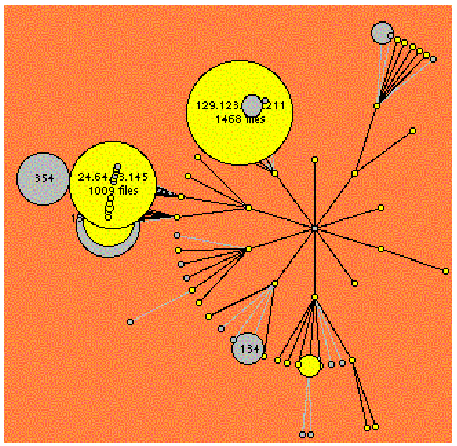
Topologia della rete di overlay

- *centralizzata*
- *parzialmente centralizzata (ibrida)*
- *puramente peer-to-peer*

Stato dell'arte:
dynamic supernode overlay networks.



Organizzazione logica della rete di overlay

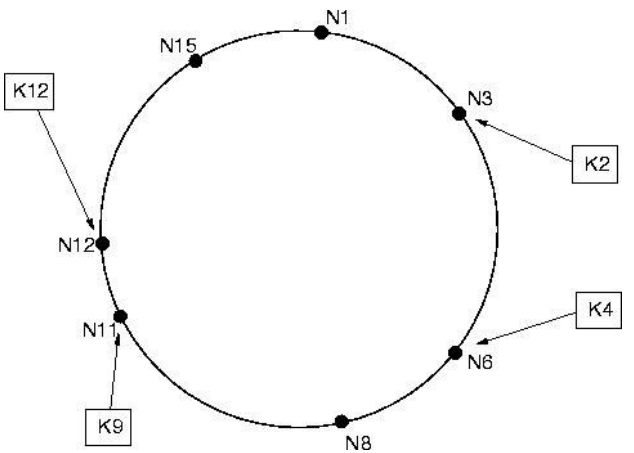


Le reti di overlay *unstructured* non sono vincolate da una organizzazione logica deterministica dei nodi.

es. Gnutella, Freenet, JXTA

Le reti di overlay structured sono caratterizzate da una organizzazione deterministica delle connessioni tra i nodi.
L'algoritmo di routing dei messaggi è strettamente legato a tale organizzazione.

es. Chord



Centralized Directory Model (CDM)

I peer si connettono a un direttorio centrale (virtuale!) per pubblicare informazioni sui contenuti che offrono in condivisione.

Quando un peer invia una query al direttorio, quest'ultimo risponde fornendo un elenco (ordinato secondo un qualche criterio) dei peer che condividono l'oggetto richiesto.

Lo scambio di dati avviene poi da peer a peer.

Vantaggi: semplice, dà controllo sui contenuti condivisi
Limiti: poco scalabile, e il direttorio richiede una implementazione robusta

Es.: Napster, BitTorrent

Napster

Architettura: centralizzata, unstructured
Modello: CDM

Il server di Napster non ospita i file, che risiedono sui PC degli utenti. Lo scopo del server Napster è quello di implementare il direttorio CDM e quindi di mettere in contatto i peer.

Napster 2.0 offre servizi a pagamento per accedere al più ampio catalogo di musica online.
http://www.napster.com

BitTorrent

Architettura: centralizzata, unstructured
Modello: CDM

Il server (accessibile via Web) non contiene informazioni sulla locazione dei file, ma fornisce i file *.torrent* che li descrivono (nome, lunghezza, ecc.) e li associano all'URL di un *tracker*.

I tracker, che utilizzano un semplice protocollo costruito su HTTP, aiutano i downloader a entrare in contatto tra loro. I downloader inviano informazioni sul proprio stato ai tracker, che rispondono con liste di peer che stanno scaricando lo stesso oggetto.

I file vengono spezzettati in parti e poi in ulteriori sottoparti, che vengono scambiate secondo uno schema "rarest first".

http://bittorrent.com

Flooded Requests Model (FRM)

Quando un peer genera una query, la propaga a tutti i suoi vicini, indistintamente, i quali fanno lo stesso. Il numero massimo di salti che una query può fare è un parametro fissato (TTL, time to live).

$$TTL(0) = TTL(i) + Hops(i)$$

Vantaggi: efficiente in comunità di dimensioni limitate
Limiti: richiede molte risorse in termini di banda

Es.: Gnutella

Gnutella

http://rfc-gnutella.sourceforge.net

Architettura: pura, unstructured
Modello: FRM

I nodi si connettono alla rete tramite nodi stabili e noti (questa fase non fa comunque parte del protocollo).

I messaggi consentiti da Gnutella possono essere così raggruppati:

Group Membership (`PING` e `PONG`, per la ricerca dei peer)
Search (`QUERY` e `QUERY HIT`, per la ricerca dei file)
File Transfer (`GET` e `PUSH`, per lo scambio di file tra peer)

Per evitare di congestionare la rete, i messaggi `PING` e `QUERY` sono sempre associati a un TTL (in genere pari a 7).

Gnutella - (2)

Nella rete Gnutella reale è stato osservato che l'alto costo per il broadcast e la scarsità di risorse (dovuta all'abbondanza di free rider) porta alla frammentazione della stessa rete in più sottoreti.

Distribuzione del grado di nodo in Gnutella: *multi-model, combina una legge di potenza (per i nodi con più di 10 link) e una distribuzione quasi-costante (per i nodi con meno di 10 link)*.

$$P(k) \sim k^{-\tau}$$

Internet: *grado di nodo distribuito secondo una legge di potenza*.

La non perfetta sovrapposizione delle due topologie è causa di
- uso inefficiente della struttura di rete fisica
- mancanza di robustezza

ma anche di
- buona tolleranza ai guasti

Document Routing Model (DRM)

La rete di overlay realizza una Distributed Hash Table (DHT), cioè un sistema distribuito che suddivide equamente tra i partecipanti la responsabilità delle chiavi (ciascuna delle quali è associata a un valore, che nella pratica corrispande a un file, a un blocco di dati, ecc.).

Un messaggio di ricerca contenente una chiave viene propagato all'unico responsabile di quella chiave, che fornisce in risposta il valore cercato.

La chiave viene solitamente generata applicando una funzione (detta funzione di hash) al dato stesso che deve essere inserito/cercato nella DHT.

Sono spesso utilizzate le funzioni della famiglia SHA (Secure Hash Algorithm). Applicando SHA-1 a un dato di lunghezza < 2^{64} bit, si ottiene un *digest* di lunghezza pari a 160 bit.

Document Routing Model (DRM)

indirizzo IP del peer → *peer ID*
descrizione di una risorsa → *resource ID* (chiave di ricerca nella DHT)

Pubblicazione e Replica: la responsabilità di una chiave viene data al vicino il cui peer ID è più simile alla chiave stessa; ripetere fino a quando il peer ID più simile alla chiave è quello del peer corrente

Ricerca: la query (per chiave) viene inviata al vicino il cui peer ID è più simile alla chiave cercata; il processo si ripete fino al ritrovamento della chiave cercata, oppure fino a quando il peer ID più simile alla chiave è quello del peer corrente, o fino all'esaurimento del TTL.

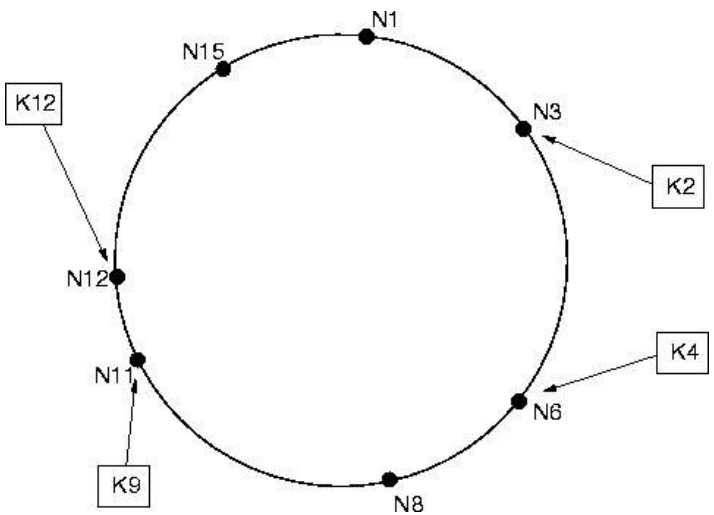
Vantaggi: scalabilità
Limiti: protocolli complessi, difficile gestire ricerche fuzzy

Es.: Chord, FreeNet, Overnet

Chord

Architettura: pura, structured
Modello: DRM

Una funzione di hash (tipo SHA-1) assegna a ciascun nodo un identificatore a *m* bit, così come ogni oggetto è identificato da una chiave a *m* bit.
Gli identificatori dei nodi sono ordinati in un cerchio modulo 2^m.



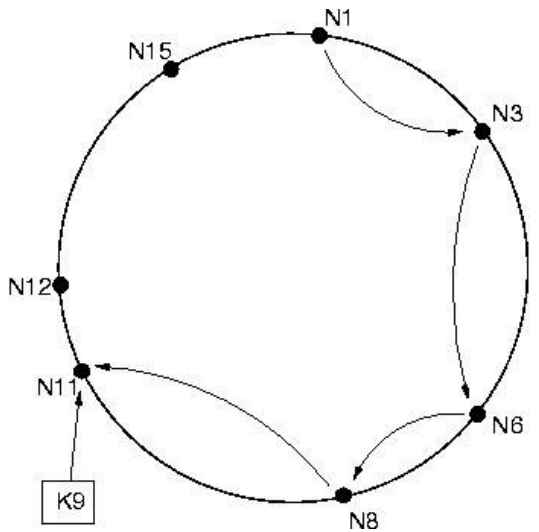
La chiave *k* è assegnata al primo nodo in senso orario il cui identificatore è ≥ *k*.

Tale nodo è detto il *nodo successore* della chiave *k*.

Chord - (2)

Algoritmo di base per la ricerca:

Ciascun nodo conosce il nodo con l'identificatore immediatamente superiore. Una query per una certa chiave viene propagata in senso orario fino al primo nodo il cui identificatore è ≥ alla chiave.



La risposta alla query fa il cammino inverso, fino all'originatore della query.

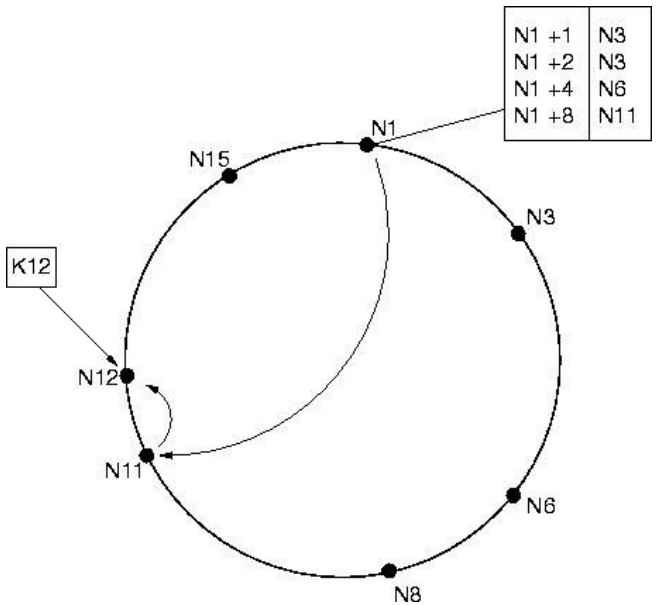
Seguendo questo approccio, il numero di nodi da attraversare se la rete è composta da *N* nodi è

O(N)

Chord - (3)

Algoritmo di ricerca accelerato:

Ciascun nodo *n* mantiene una routing table con un massimo di *m* entry, detta *finger table*.
La *i*-esima entry è: $n.finger[i] = successor(n + 2^{i-1})$



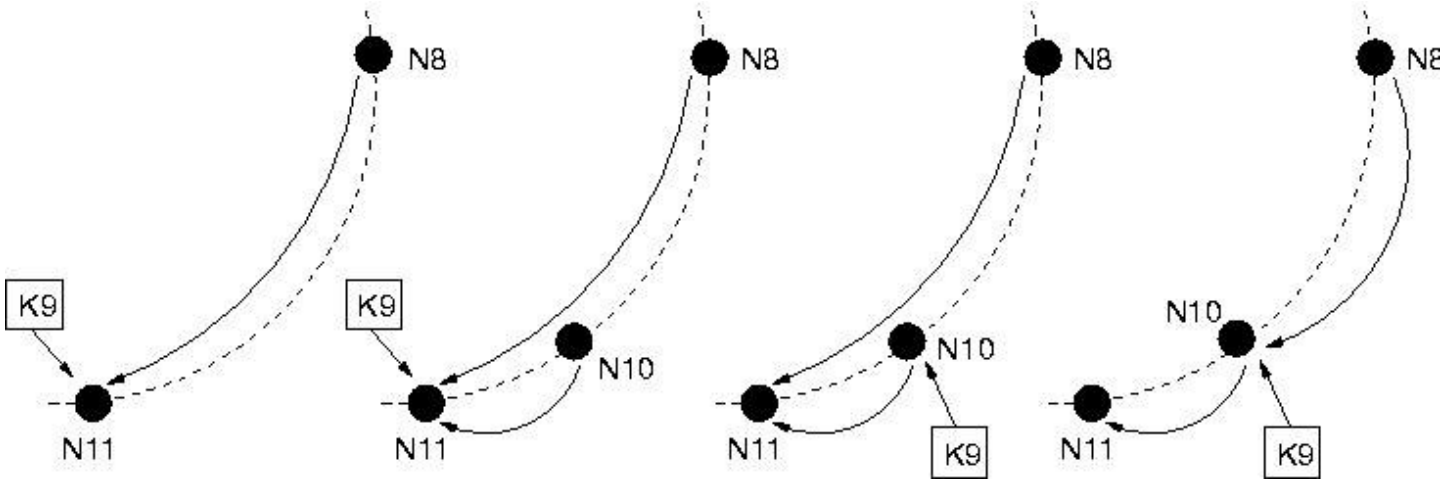
Utilizzando la finger table per trovare il nodo il cui identificatore precede quello del nodo successore della chiave cercata, il numero di nodi da attraversare se la rete è composta da *N* nodi è

$$O(logN)$$

con elevata probabilità.

Chord - (4)

Per assicurare una corretta esecuzione dell'algoritmo di ricerca anche quando l'insieme dei nodi della rete cambia, Chord prevede un protocollo di stabilizzazione che ciascun nodo dovrebbe eseguire periodicamente in background, per aggiornare la finger table.



Chord - (5)

La correttezza del protocollo Chord si basa sul fatto che ciascun nodo conosce almeno il suo successore. Purtroppo, può succedere che un nodo si disconnetta inavvertitamente (senza seguire la procedura dettata dal protocollo).

Per una maggiore robustezza, ciascun nodo Chord manitene una *successor list* che contiene i puntatori ai primi *r* successori del nodo stesso.

FreeNet

Architettura: pura, unstructured
Modello: DRM

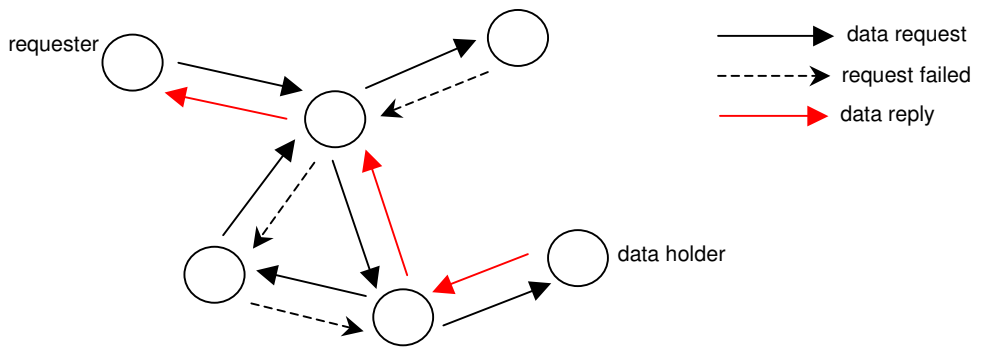
Principali obbiettivi:

- mantenere la privacy per chi produce e consuma informazioni
- resistere alla censura
- garantire disponibilità e affidabilità grazie alla decentralizzazione
- fornire un sistema scalabile

Ciascun file condiviso ha una chiave associata, tipicamente generata con SHA-1 a partire dalla descrizione testuale del file.
Ciascun peer è inizialmente l'unico responsabile delle proprie chiavi.

Ciascun peer mantiene una routing table che contiene gli indirizzi di altri nodi e per ciascuno un elenco di chiavi presenti con alta probabilità (in base alla "storia passata").

FreeNet - (2)



Quando un nodo riceve una query, controlla se la chiave cercata è in suo possesso e in caso affermativo invia il file al peer da cui ha ricevuto la query.

Altrimenti il peer propaga la richiesta al nodo che più probabilmente possiede la chiave cercata (in base alla routing table). C'è un TTL, ovviamente.

I file trovati vengono propagati seguendo il percorso inverso della query. Questo mantiene l'anonimato tra consumatore e produttore.

Durante la fase di upstream vengono aggiornate le routing table e il file (con relativa chiave) può anche essere salvato nella cache dei nodi intermedi. Quest'ultima operazione risponde a un criterio di *clustering* di chiavi simili.

Distributed Indexes and Repositories Model (DIRM)

Alcuni peer hanno (detti *broker*) che indicizzano dinamicamente un certo numero di peer locali, e in certi casi replicano parzialmente gli indici dei broker con cui sono connessi.

Le query di un peer vengono propagate al suo broker di riferimento e ad altri broker ancora, che forniscono elenchi di contenuti rispondenti alle richieste, assieme ai puntatori ai peer che fisicamente posseggono quei contenuti.

Vantaggi: scalabilità.
Limiti: per mantenere un buon livello di consistenza tra gli indici dei broker servono protocolli pesanti.

Es. OpenNap, Direct Connect, eDonkey, eMule

OpenNap

Architettura: ibrida, unstructured
Modello: DIRM

E' l'evoluzione di Napster che supporta la comunicazione tra server che indicizzano i contenuti dei peer.

WinMX era il client OpenNap più famoso.

Direct Connect

Architettura: ibrida, unstructured
Modello: DIRM

La rete di overlay è composta dagli Hub, dai Client dal HubListServer.

Gli Hub forniscono un servizio di naming e servono a mettere in contatto diversi Client (scambio di contenuti, messaggistica istantanea).

I Client trovano gli Hub grazie al HubListServer.

http://www.dslreports.com/faq/dc

eDonkey, eMule

Architettura: ibrida, unstructured
Modello: DIRM

Sono le più famose reti peer-to-peer basate su DIRM.
Ciascun peer pubblica informazioni relative ai propri contenuti su server che agiscono da broker. Chiunque può mettere a disposizione un server.
Recentemente nel client ufficiale di eDonkey è stato introdotto il supporto per Overnet (che è di tipo DRM).

http://www.edonkey2000.com
http://www.emule-project.net

Selective Queries Model (SQM)

I peer con più capacità di banda, potenza di calcolo e storage operano da *supernodi*, assumendo la responsabilità di propagare i messaggi nella rete di overlay. I peer meno "dotati" possono pubblicare e fare ricerche, ma non contribuiscono al processo di routing

Vantaggi: scalabilità.
Limiti: peer malevoli possono insinuarsi tra i supernodi e compromettere parzialmente l'efficienza del sistema.

Es. FastTrack, Gnutella 2, JXTA

FastTrack

Architettura: ibrida, unstructured
Modello: SQM

Un peer ospitato da una macchina dotata di buone capacità (valutate dinamicamente) diventa supernodo.

I supernodi sono tra loro connessi e si scambiano informazioni per soddisfare le richieste dei propri "leaf" peer.

KaZaA Media Desktop: http://www.kazaa.com
Skype: http://www.skype.com

JXTA

Architettura: ibrida, unstructured
Modello: SQM

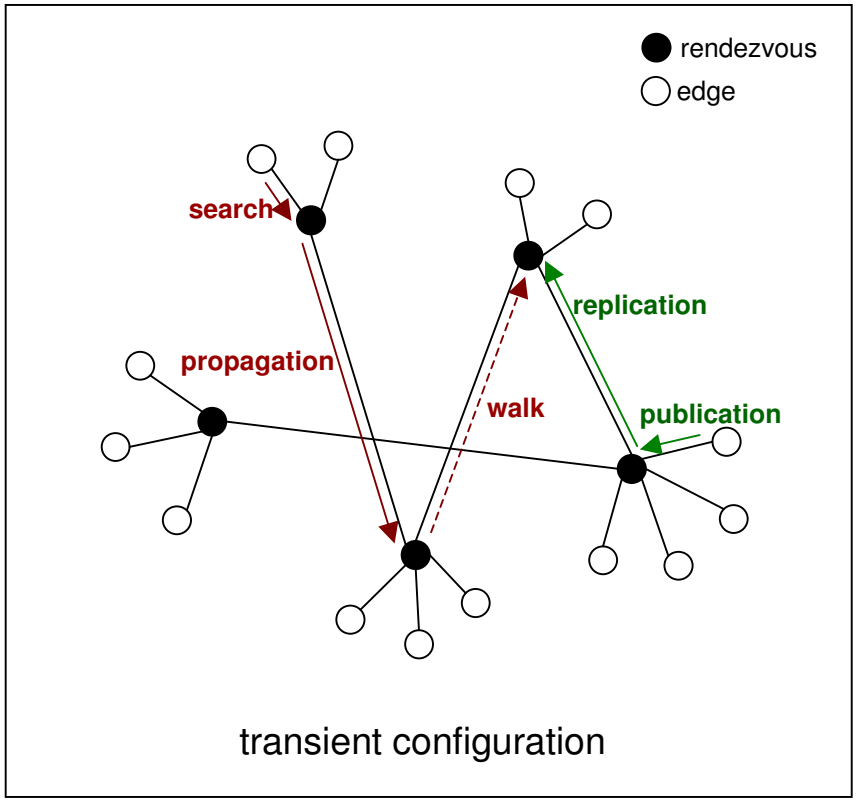
Un peer ospitato da una macchina dotata di buone capacità (valutate dinamicamente), diventa supernodo (*rendezvous super-peer*), a cui sono connessi altri peer meno performanti (*edge peer*).

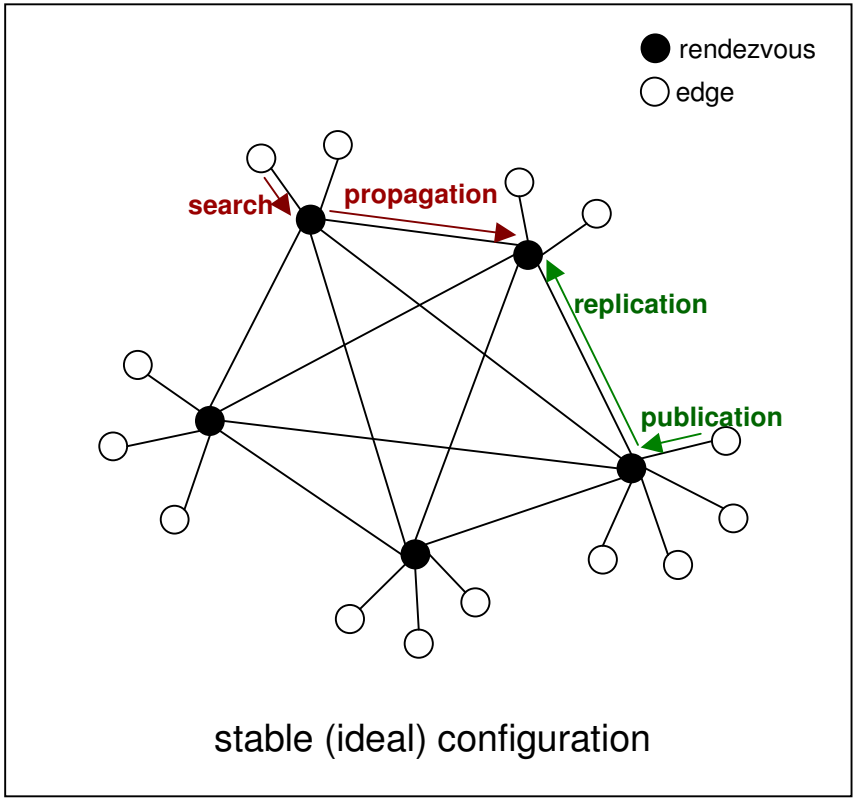
I supernodi sono tra loro connessi e propagano i messaggi secondo utilizzando una funzione di hash, e aggiustando il tiro con un *limited range walker*.

http://www.jxta.org

JXTA - (2)

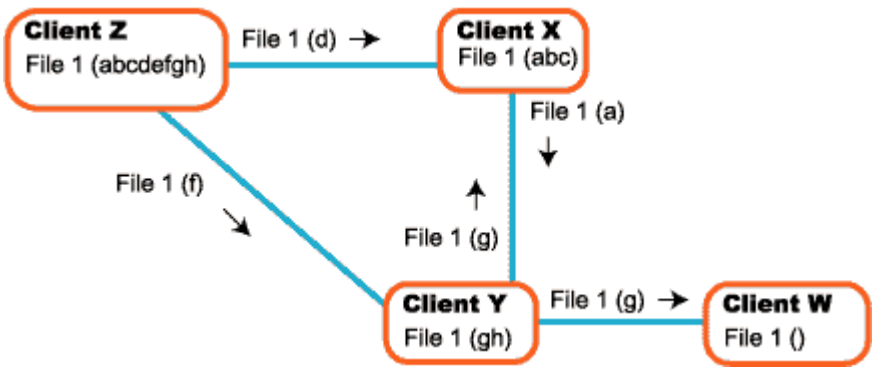
Coppie (attributo,valore) vengono estratte dagli *advertisement* da pubblicare.





Appendice: Multisource File Transfer Protocol (MFTP)

MFTP è stato progettato in modo da permettere la distribuzione di file nel modo più rapido possibile tra tutti i peer interessati. Questi ultimi scaricano i file prendendone pezzi diversi da sorgenti diverse, e diventando a loro volta fornitori delle parti già scaricate.



Diverse reti peer-to-peer utilizzano MFTP per il trasferimento dei file: eDonkey, eMule, Overnet sono gli esempi più famosi.

Skype

Introduzione a Skype

Tipologia applicativa: *servizio di VoIP gratuito*

Autori: *N. Zennstromm and J. Friis (già ideatori di KaZaA)*

Supporto offerto: *Windows, Mac OS X, Linux, Pocket PC*

Caratteristiche distintive:

- *si basa sul modello peer-to-peer SQM*
  (unico elemento centralizzato: il Login Server)
- *permette di attraversare firewall e NAT, grazie ai supernodi*
  (ciascun peer usa STUN e TURN per capire se è dietro NAT)
- *offre qualità della voce migliore rispetto alla concorrenza*
  (trasmissione su UDP a 36Kbps; codec 50-8K Hz di Global IP Sound)
- *Implementa noti standard crittografici*

Registrazione

L'utente decide il proprio username A, e una password P_A.

A partire da (A,P_A) il peer utente genera una coppia di chiavi RSA (S_A,V_A), con S_A chiave privata e V_A chiave pubblica.

Il peer utente invia A e V_A al Login Server (la cui chiave pubblica V_S è nota a tutti i peer). Il Login Server controlla che A non sia un username già esistente, e se il controllo viene superato crea un *Identity Certificate* IC_A, che contiene tra l'altro $(A,V_A)^S$, e lo invia al peer utente A.

In questa fase il Login Server si comporta da *certification authority*, fornendo al peer utente un certificato che verrà da quel momento sempre utilizzato per l'autenticazione del peer utente con gli altri peer e con il Login Server.

Login successivi al primo

Il peer utente si autentica al Login Server, che gli fornisce una lista di supernodi (questa lista è frequentemente aggiornata).

I supernodi sono peer con IP pubblico e CPU, memoria e banda sufficienti a servire qualche migliaia di peer "leaf".

Il peer utente sceglie un supernodo e apre una connessione TCP con quest'ultimo.

Il peer utente segnala la propria presenza ai membri della sua *buddy list*.
La ricerca di altri peer si basa su uno scambio di messaggi tra i supernodi.

Il peer utente determina il tipo di NAT dietro il quale eventualmente si trova.

Chiamata e comunicazione

I segnali di chiamata e di chiusura della comunicazione avvengono sempre via TCP (direttamente tra peer A e peer B oppure tramite uno o più supernodi).

Peer A e peer B utilizzano il seguente *key agreement protocol*:

1) scambio di pacchetti da 64 bit, firmati con chiave privata e verificati dal ricevente usando la chiave pubblica del mittente.

2) mutua autenticazione basata sullo scambio degli IC

3) creazione della chiave di sessione SK_{AB} a 256 bit (ciascun peer contribuisce con 128 bit)

La comunicazione avviene su UDP, usando AES con la chiave di sessione SK_{AB}. In tal modo anche se la comunicazione avviene tramite supernodi, solo mittente e destinatario possono decodificare i pacchetti.

Riferimenti bibliografici

www.skype.com

Skype reverse-engineered, di Frank Bulk (maggio 2004)

Skype security evaluation, di Tom Berson (ottobre 2005)